J. Automation & Systems Engineering 4-4 (2010): 220-233

Regular paper

JASE
Journal of Automation & Systems Engineering

Decentralized Neural-Fuzzy Control of a Six Links Robot Provided with DC Motors

R. Mellah and S. Guermah and R. Toumi

In this paper, a stable fuzzy adaptive controller for trajectory tracking is developed for a six link PUMA560 robot manipulator provided with DC motors. The control scheme consists of two main control loops. The first loop requires three controllers: neural-fuzzy controller I, NN controller II, and a sliding mode controller. Neural-fuzzy controller I is a Tkagi-Sugeno fuzzy controller with adaptive neuro-fuzzy inference system (ANFIS) structure, which can improve its performance and achieves satisfactory results through online learning. The NN controller II, is used to drive manipulator joints to compensate for the errors caused by the unstructured and structured uncertainties while controlling the robot manipulator. The sliding mode controller, is used to make the tracking errors approach zero. As for the second loop calculates the DC motors voltages by realizing a torque feedback.

The simulation results obtained showed that the proposed controller has achieved a high tracking performance. In addition the results have proved the effectiveness and robustness of the proposed controller to achieve considerable enhancement of the motion tracking a robot characterized by unknown disturbances and parameter uncertainties.

Keywords: Feedback control, Neural-fuzzy Systems, Adaptive Control, Manipulator robot, Sliding control.

1. INTRODUCTION

Robots are nowadays one of the most important pieces of machinery for industrial automation, normally used to perform generic tasks such as pick and place, seam tracking, assembly etc [1]. Therefore design of robust adaptive controllers suitable for real-time control of robot manipulators is one of the most challenging tasks for many control engineers, especially when manipulators are required to manoeuvre very quickly under external disturbances [2]. In the literature, a vast collection of control approaches are available for manipulating robot.

Historically, the first approach attempted of the nonlinear systems was based on the linearization around a nominal trajectory [3]. This method is applied to various fields of study: asymptotic stability, and local observability, thus identification in the vicinity of an operating point. Nevertheless, although its effectiveness is recognized in particular situations, it presents major defaults which condemn it as a general tool to use. [4]

- The linearization of a non linear system gives only a very partial description of the input-output behaviour.
- In the case of system having unlimited nonlinearities the linearization is not robust.

Laboratoire de conception et de conduite des systèmes de production (L2CSP), Faculté de Génie Electrique et d'Informatique, Université Mouloud Mammeri Tizi.Ouzou (U.M.M.T.O). - BP n° 15000 – Tizi-Ouzou (Algeria)
Robotic Laboratory Genie Electric Faculty, U.S.T.H.B – BP n° 32 – El Alia – Bab Ezzouar –16111 – Alger Algeria.

- Affination per pieces which approximates locally a nonlinear system can have input-output behaviour qualitatively different from the nonlinear system one.

In the last few decades, much research effort have been put into the design of intelligent controllers using fuzzy logic, because fuzzy control is a model-free adaptive control strategies [2]. It offers capabilities of dealing with uncertainties, which cannot be described by precise mathematical models. Fuzzy control is a potentially powerful approach that can capture human experience and expertise in controlling complex processes, thereby circumventing many of the shortcomings of hard-algorithmic control [5]. Since fuzzy control is a model - free approach, it may provide a possible solution to robotic control, which deals with high nonlinearity, high coupling, and unmodelled uncertainties. Fuzzy control has been applied to real control systems, although conventional fuzzy systems present some inconvenience in design, such as finding exact membership functions and determining suitable fuzzy control rules [6]. On the other hand, much research effort has been put into the design of neural networks (NNs) applications in both continuous and discrete time for robotic control. Recently, hybrid control laws containing NNs have attracted more and more attention because NNs were used to adjust and optimize parameters of fuzzy controllers through offline or online learning [6].

In our case, the dynamic nonlinear model is a six-degree freedom polar of PUMA560 robot manipulator equipped with D.C. motors. It is a multivariable system, which is nonlinear and strongly coupled and is frequently subjected to structured and/or unstructured uncertainties even in a well-structured setting for industrial use. Structured uncertainties are mainly caused by imprecision in the manipulator link properties, unknown loads, and so on. Unstructured uncertainties are caused by unmodeled dynamics, e.g., nonlinear friction, disturbances, and high-frequency models of the dynamics. Thus the influence of the disturbances and the dynamic interaction between the actuators and the articulations of the manipulator arm, are not always easy to estimate and possibly vary with time. Consequently, it is difficult to obtain an accurate mathematical model, so that computed torque controllers or other model-based controllers could be accurately applied. Although, adaptive control methods can achieve fine control and compensate for partially unknown manipulator dynamics, they often suffer from heavy computational burden and this hinders their real-time applications [2]. Furthermore, when tracking precision is strictly required under high uncertainty, control performance is often degraded due to the existence of modelling errors and external disturbances.

On the other hand, the good tracking performance can be realized by control systems using the nonlinear design methods based on Neural-fuzzy structures [5]. Therefore, a fuzzy controller is able to supply comparatively good control performance for a nonlinear, multi-input, multi output coupled system, which is a typical robotic situation. However, there are many difficulties in determining the membership functions for each input and output. Moreover, since the dynamics of the robotic manipulator involve many changing factors such as structured and unstructured uncertainties, there is no satisfactory solution unless the membership functions can be automatically adjusted [6], in a timely way, to compensate for the effects of these factors. That is why the structure of neural networks provides powerful abilities, such as learning abilities, optimization abilities, and connectionist structures and generalization, to the fuzzy controller [2]. Thus, some researchers have tried to combine neural networks with fuzzy logic to design control schemes for robot manipulator. In our proposed control scheme neural networks are used to adjust and optimize online parameters of fuzzy controllers with learning algorithm in order to approximate the nonlinear components in the dynamic system of robot manipulator, and Lyapunov stability theory is employed to design a closed-loop control system with stability convergence, and improved robustness. Among this scheme adapted to the dynamic control

of a manipulator arm, we opted for Neural-fuzzy control by state feedback. The latter is planned starting from the model of the system to be designed and is used in order to distinguish the stage actuators from the mechanical stage [7]. The first stage permits to obtain the desired torque applied to each joint, by a Neural-fuzzy structure and the second one calculates the voltage to be applied to the motors through a torque feedback.

The remainder of the paper is organized as follows. Section 2 introduces the dynamic model of PUMA560 robot. In section 3 we clarify the overall strategy of synthesised Neural-fuzzy control. The simulation results are presented in section 4, to illustrate the tracking performance of the proposed control scheme. Finally, in section 5, some conclusions are presented.

2. DYNAMIC MODEL OF PUMA 560 ROBOT PROVIDED WITH D.C. MOTORS

The combination of the motion of a six-degree of freedom mechanical system defined by the generalized coordinates $q \in R^6$, of PUMA 560 robot manipulator given by [8] with the equations of the D.C. motors allows to have the compact form of the dynamic model of PUMA 560 driven by D.C. motors. The equation of motion can be written as:

$$\begin{aligned} \tau &= M(q)\ddot{q} + B(q)\left[\dot{q}\,\dot{q}\right] + C(q)\left[\dot{q}^2\right] + G(q) \\ U_{mj} &= Ri_j + L\frac{di_j}{dt} + K_{ej}\frac{dq_{mj}}{dt} \end{aligned} \tag{1}$$

Where $q, \dot{q}, \ddot{q} \in \mathrm{R}^6$ are the vectors of joint displacements, velocities, and accelerations, respectively.

$\tau \in R^6$ is a vector of generalized joint torques.

$M(q) \in R^{6*6}$ is the symmetric positive definite manipulator inertia matrix.

$B(q) \in R^{6*15}$ contains coriolis terms.

$C(q) \in R^{6*6}$ contains centripetal terms.

$G(q) \in R^6$ represents gravitational effects.

q_{mj} is the angular position of the rotor.

R_j is the resistance of the armature circuit.

L_j is the inductance of the armature circuit.

I_j is the armature current.

K_{ej} is the voltage constant of the motor.

U_{mj} is the armature voltage.

3. INTELLIGENT NEURAL-FUZZY CONTROL ARCHITECTURE

Recently, neural-fuzzy control has become a powerful tool in control engineering, especially in systems that are structurally difficult to model to their inherent natural nonlinearities and other modelling complexities such as robotic manipulator. Indeed neural-fuzzy control can potentially overcome all the kinds of difficulties in precision robotic control experienced by conventional control methods.

Fig.1 shows the structure of the proposed overall control system. The proposed control system for each manipulator joint consists of three major controllers: neural-fuzzy controller I, neural-fuzzy controller II and a sliding-mode controller.

Neural-fuzzy controller I is used to force the system outputs defined by the vectors of joint displacements q, and velocities $\dot{q}$, to track the desired trajectories, respectively, in order to achieve performance of robotic manipulator with satisfactory results through online learning. The neural-fuzzy controller II is used to drive the manipulator joints in order to compensate for the errors caused by the unstructured and structured uncertainties while controlling the robot manipulator. The errors between the joint desired and present position / velocity values are used to train the neural-fuzzy controller II online. The learning algorithm guarantees that the closed- loop system will be stable. The sliding-mode controller, $\mathrm{sgn}\left(C_{dot}\,\dot{e} + C\,e\right)$, is used to make the tracking errors approach zero, where e is the error between the robot joint desired position q_d and its present position q, $\dot{e}$ is the error between the robot joint desired velocity $\dot{q}_d$ and its present velocity $\dot{q}$, and where C_{dot} and C (constants) are the contributions of $\dot{e}$ and e to the overall performance objective, respectively.

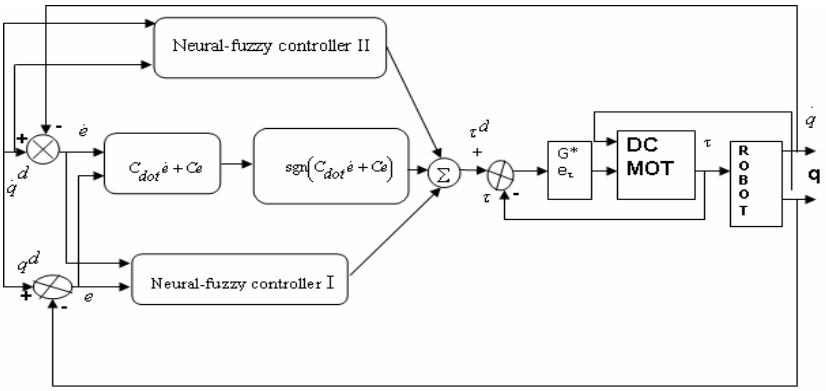

Fig.1. The overall neural-fuzzy control system

3.1. Neural -Fuzzy Controller I

The neural-fuzzy controller I is a Takagi-Sugeno fuzzy controller with adaptive neural-fuzzy inference system (ANFIS) architecture illustrated by fig. 2 [9]. The ANFIS structure is a multilayer whose connections are not pondered, or have all one weights of 1. The neurons are of two different types according to their functionality: the square neuron (adaptive) contains some parameters, and the circular neuron (stationary) doesn't have any

parameters. However, every neuron (square or circular) applies a function on its input signals.

The neural-fuzzy controller I develop here has two inputs i.e. e and $\dot{e}$, and single output i.e. y, where e is the error between joint desired and actual positions, $\dot{e}$ is the error between joint desired and actual velocities; y is the controller I's output (torque). These two input variables considerably affect the control action. In our simulation, we have six joints.

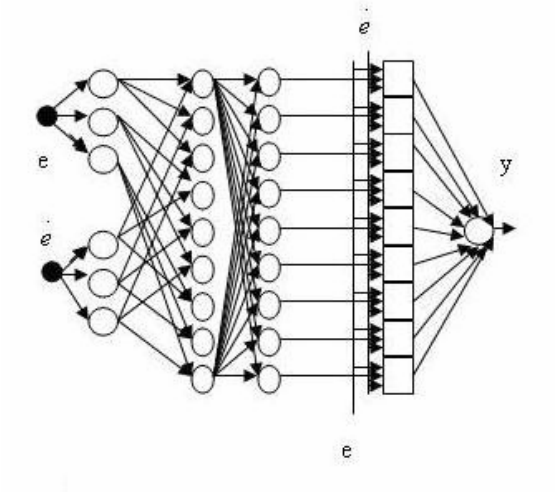

Fig 2 Structure of the neural-fuzzy controller I

3.1.1. Control algorithm

The function of the node at the first layer is identical to the membership function in the fuzzification process.

$$O_{1,i} = \begin{cases} \mu_{Ai} & for\ i = 1,3 \\ \mu_{B\,i-3} & for\ i = 4,6 \end{cases} \tag{2}$$

Where μ_A and μ_B are triangular functions with fixed parameters associated with the premises. Thus in order to limit the complexity of the structure of the regulator, we have considered only three sets, noted M^-, M^0 and M^+ representing the fields in which the errors are either negative, or null, or positive. Thus, each rule depends on three premises and gives a conclusion in the form of a law noted u_j, accompanied by its degree of membership.

Each node of the second layer is a circular node named Π who generates in output the product of its input. This product represents the degree of activation of rule in such a way that:

$$O_{2,1} = O_{1,1} * O_{1,4} = W_1$$
$$O_{2,2} = O_{1,1} * O_{1,5} = W_2$$
$$O_{2,3} = O_{1,1} * O_{1,6} = W_3$$

$$
\begin{aligned}
O_{2,4} &= O_{1,2} * O_{1,4} = W_4 \\
O_{2,5} &= O_{1,2} * O_{1,5} = W_5 \\
O_{2,6} &= O_{1,2} * O_{1,6} = W_6 \\
O_{2,7} &= O_{1,3} * O_{1,4} = W_7 \\
O_{2,8} &= O_{1,3} * O_{1,5} = W_8 \\
O_{2,9} &= O_{1,3} * O_{1.6} = W_9
\end{aligned}
\tag{3}
$$

Each node of the third layer calculate the standardization of the numerical data, using the following expression:

$$O_{3,i} = \bar{W}_i = \frac{W_i}{\sum_{k=1}^{9} W_k} \tag{4}$$

Each node of the fourth layer is a squared node with a function described as follows:

$$O_{4,i} = \bar{W}_i \left(a_i e + b_i \dot{e} + r_i \right) \tag{5}$$

Where a_i , b_i and r_i are parameters to be adjusted.

Finally, the fifth layer contains only one node which calculates the sum of the outputs of the nodes of the preceding layer:

$$O_5 = gu * \sum_{i=1}^{9} O_{4,i}. \tag{6}$$

Where gu is the adaptation gain.

3.1.2. Training algorithm

The basic idea of our work consists in calculating the desired couple to be applied to the manipulator joints, by a neural-fuzzy system whose unknown parameters are estimated by a learning algorithm based on the wide band filter of Kalman which consists in linearizing the output y at any moment around the vector $\hat{\theta}_1$ estimated by minimizing each joint angle error defined as:

$$e_i(t) = q_i(t) - q_i^{\,d}(t) \tag{7}$$

The adaptation law used is given by the following relation: [9]

$$\theta_{1i}(t+1) = \theta_{1i}(t) + P(t) J_{\theta_{1i}} e(t+1) * Te \tag{8}$$

Where Te is the sampling period, $P(t)$ is the adaptation profit given by: [9]

$$P(t) = \frac{1}{1 + J^T_{\theta_{1i}} J_{\theta_{1i}}} \tag{9}$$

Where $J_{\theta_{1i}} = \dfrac{\partial q_i}{\partial \theta_{1i}}$ is calculated retro-propagation method.

3.2. Neural -Fuzzy Controller II

This controller uses a neural-fuzzy inference (NFI) which realizes the process of fuzzy reasoning using the structure of neural networks (NN) and expresses the parameters of fuzzy reasoning through the weights of a NN. NFI can automatically identify the fuzzy rules and tune the membership functions by modifying the connection weights of the NNs using a self-learning algorithm. The membership functions in NFI will be finally optimized by the NN structure.

Suppose that a fuzzy system consists in L fuzzy rules, each of which has N input variables, $x_1, x_2, ..., x_N$. Then the l^{th} rule $(R^{(l)})$ is expressed as

$$R^{(l)}: \textit{If } x_1 \textit{ is } F_1^l \textit{ and .... and } x_N \textit{ is } F_N^l \textit{ then } y_1^l \textit{ is } H_1^l,\ y_2^l \textit{ is } H_2^l, ..., y_M^l \textit{ is } H_M^l.$$

By using singleton fuzzification and inference product, the j^{th} fuzzy output for the l^{th} fuzzy rule, $y_j^l, \left(j = 1,2,.....,M\right)$, can be expressed in this form: [6]

$$y_j^l = \overline{y}_j^l * \left[\prod_{i=1}^{N} \mu_{F_i^l}(x_i)\right]. \tag{10}$$

where $\overline{y}_j^l$ is the point that has the maximum fuzzy membership value to fuzzy set H_j^l, namely, $\mu_{H_j^l}\left(\overline{y}_j^l\right)=1$, and $\mu_{F_i^l}(x_i)$ is the fuzzy membership value of the i^{th} input of the l^{th} fuzzy rule.

By taking the center-average defuzzifier mapping, the crisp value of the j^{th} output y_j is derived as [6]

$$y_j = \frac{\sum_{l=1}^{L} \overline{y}_j^l * \left[\prod_{i=1}^{N} \mu_{F_i^l}(x_i)\right]}{\sum_{l=1}^{L}\left[\prod_{i=1}^{N} \mu_{F_i^l}(x_i)\right]} \tag{11}$$

If we define the following equation as the nonlinear mapping of an NN:

$$G_l(x) = \frac{\prod_{i=1}^{N} \mu_{F_i^l}(x_i)}{\sum_{l=1}^{L}\left[\prod_{i=1}^{N} \mu_{F_i^l}(x_i)\right]} \tag{12}$$

And $\overline{y}_j^l$ is an adjustable weight of the NN that also can be written as W_l, the fuzzy inference can be described in the form of L-E (linear-equivalent) NN:

$$y_j = \sum_{l=1}^{L} W_l G_l(x) \tag{13}$$

where W_l is the weight associated to $G_l(x)$.

3.2.1 CONTROL ALGORITHM

We choose q_d and $\dot{q}_d$ as the desired inputs of the NN because they can be bounded reliably, whereas the system variables q and $\dot{q}$ increase or decrease rapidly during robotic control. The reason for designed neural fuzzy controller II is that its output (torque) is able to compensate for the system uncertainties. Neural-fuzzy controller II's output must reflect the system's displacement change (joint positions and velocities) and be able to drive robot

joints to reach their desired position and velocity values. Thus, the function $\left(C_{dot}\,\dot{e} + Ce\right)$ is chosen to train neural-fuzzy controller II whose structure is chosen as:

Let be $\tau_2 = G_2(x_2)^T W_2$, the neural-fuzzy controller II's output (torque) matrix (6*1), where $G_2(x_2)$ is neural-fuzzy nonlinear mapping matrix (48*6) of controller II; W_2 is adjustable weight matrix (48*1) of controller II; x_2 is the input matrix (12*1) of controller II, which can be expressed as $x_2 = \left[q_{d1}\ q_{d2} \ldots q_{d6}\ \dot{q}_{d1}\ \dot{q}_{d2}\ \ldots\ \dot{q}_{d6}\right]^T$, and where q_{di}(i=1…6) are the joints desired position, and $\dot{q}_{di}$(i=1 …6) are the joints desired velocity; and τ_2 is controller II output (torque) matrix (6x1).

The online training algorithm controller II is [6]

$$\dot{W}_2 = \Gamma_2 G_2(x_2)\left(C_{dot}\,\dot{e} + Ce\right), \tag{14}$$

where Γ_2 is a constant positive-definite matrix (48*48), called learning rate for controller II.

3.3 Sliding-mode controller

To enhance system robustness against neural-fuzzy approximation error $\varepsilon_m \operatorname{sgn}\left(C_{dot}\,\dot{e} + C\,e\right)$ is chosen as a sliding-mode controller. The magnitude of the sliding control is the bound limit value of the approximation error. Thereby, the effect of the sliding-mode controller can be made as small as possible.

The output (torque) matrix (6x1) of the sliding-mode controller is

$$\tau_3 = \varepsilon_m \operatorname{sgn}\left(C_{dot}\,\dot{e} + Ce\right) \tag{15}$$

In which ε_m is a constant chosen as small as possible.

The overall system control is concluded to be asymptotically stable in the sense of Lyapunov according to the LaSalle theorem. For more information see [6].

4. SIMULATION RESULTS

The feasibility and efficiency of the approach described in the previous sections have been studied by a series of simulations. During the simulations, the main objective to illustrate that the tracking error of the manipulating robot PUMA560, provided with D.C. motors can be made arbitrarily small in the environment of performances described by the following output desired trajectory for each joint:

$$q_i^{\,d} = \frac{\pi}{6}(1-\cos(6*t)),\ \ i=1.....6 \tag{16}$$

Figures 3 to 5, show the position and velocity tracking errors and the torque of manipulator joints, respectively. These results indicate that the proposed strategy gives good performance. It may be noted that the error profiles are quite acceptable for positions and velocities, and that the control efforts were similar and smooth. We note on one hand that the proposed approach has smaller tracking errors and on other hand evolution of applied the torque to each joint does not reach the extreme values, thus in terms of effort generated by the actuators, the applied torque to joint i is higher than that one on applied to

joint (i+1). In order to test the capacity of adaptation and robustness of the proposed approach, we have considered in our simulation that the friction and the external disturbance for each joint are assumed to be at t = 1.5 s

$$\tau_f = 38.3 * qv(i) + 18.9 * \cos(q(i)) \tag{17}$$

The answers to such a situation are given by figures 6 to 8. These results show that the resulting controller attenuates the disturbance from the base and tracks the desired joint trajectory despite the parametric perturbation of the manipulator, because the transient tracking performance is also better. It is reasonable to believe that the proposed control system will be able to control other similar non-linear processes as well.

A reason for the strong performance of the proposed controller was

- The sliding-mode controller $\varepsilon_m \operatorname{sgn}\left(C_{dot}\,\dot{e} + Ce\right)$ plays an important role in the neural-fuzzy controller. It enhances the system's robustness against the neural-fuzzy controller's approximation error. The magnitude of the sliding-mode controller's effort is the bound value of the approximation error. This has been verified by the simulation test, and the result is also consistent with the stability proof.

- The key to success in neural-fuzzy controller II is the method train the controller to compensate for the manipulator's uncertainties, namely, updating the algorithm of controller II. Since the objective of neural-fuzzy controller II is to decrease the manipulator's displacement and velocity errors by generating part of the torque used to compensate for the uncertainties, $\left(C_{dot}\,\dot{e} + Ce\right)$ is chosen to be the factor that triggers the adaptation of neural-fuzzy controller II. Meanwhile, in $\left(C_{dot}\,\dot{e} + Ce\right)$, the choice of C_{dot} and *C* is adjustable and related to the expected performance of the system.

In the neural-fuzzy controller II, Γ is defined as a constant positive definite matrix called learning rate. Thus, its choice affects the speed and accuracy of the overall control system.

Finally, this simulation also shows that the proposed approach can achieve the best control performance with favourable tracking performance.

5. CONCLUSIONS

This paper has been devoted to the presentation of a strategy based on a synthesized neural-fuzzy structure, starting from the traditional control approach by state feedback, applied to the control of a reference trajectory of the six link PUMA 650 robot manipulator provided with D.C. motors, in order to achieve good tracking performance even in presence of severe uncertainties and external disturbances such as those resulting from changing payloads. In this control approach, two objectives were considered. The first one was to calculate the desired optimal torque to be applied for each robotic joint, using the neural-fuzzy control system, which consists of tree controllers to overcome the robotic manipulator control difficulties faced by all conventional control schemes when uncertainties (e.g., friction, changing payload, time-varying, disturbances) cannot be ignored. The second one was to calculate the Um voltage to be applied to each D.C. motor, by carrying out a state feedback. The adaptation laws adopted in the parameter adjustment technique consist in projecting the fuzzy inferences rules in a connectionist network in order to adjust the parameters of the whole inference rules so that to obtain weakest position and velocity errors. The constant gain G used to refine stability.

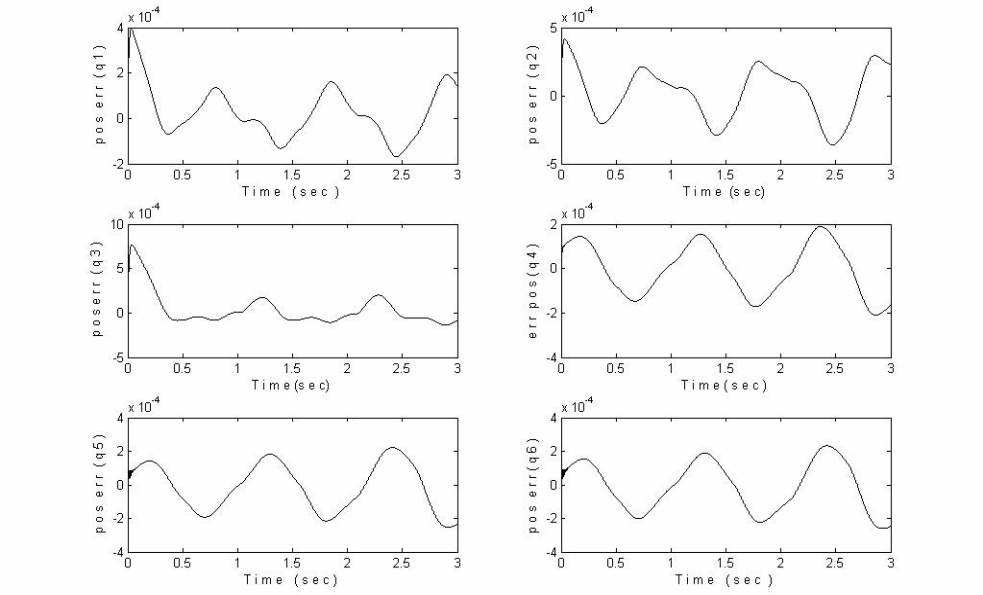

Fig. 3 Position tracking errors in the case without disturbances

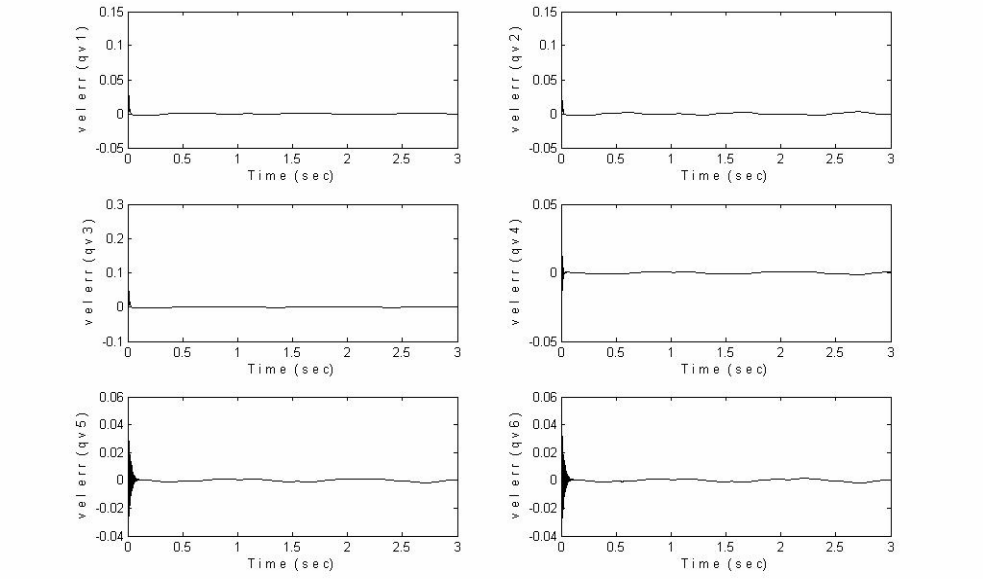

Fig 4. Velocity tracking errors in the case without disturbances.

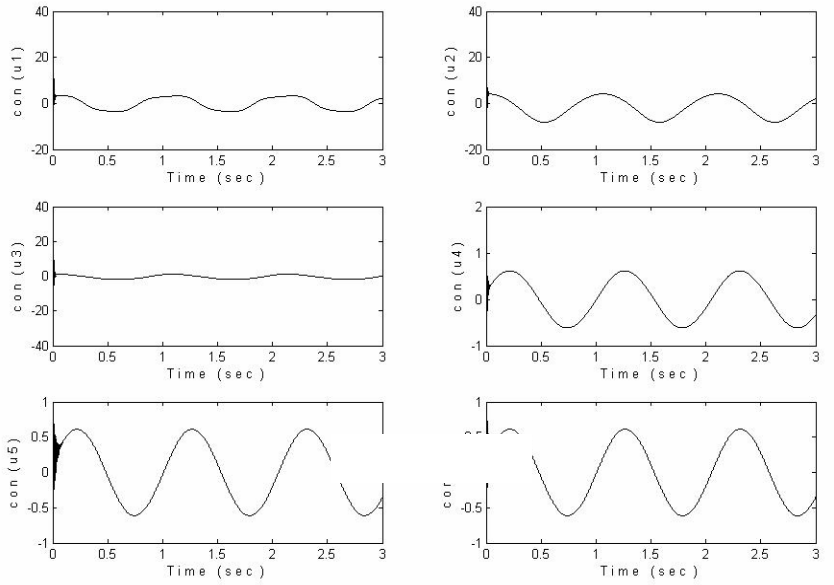

Fig 5. Profiles of the torques in the case without disturbances

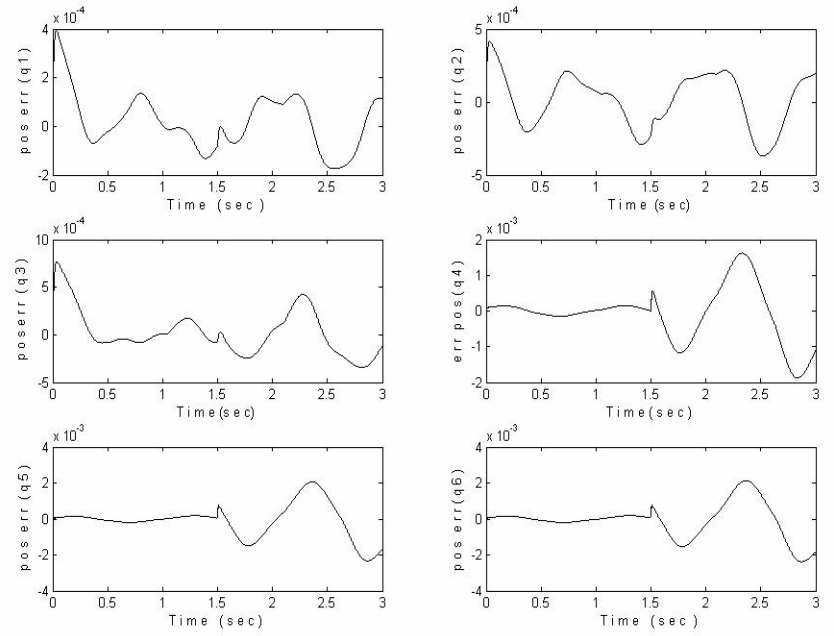

Fig 6. Position tracking errors in the case with disturbances

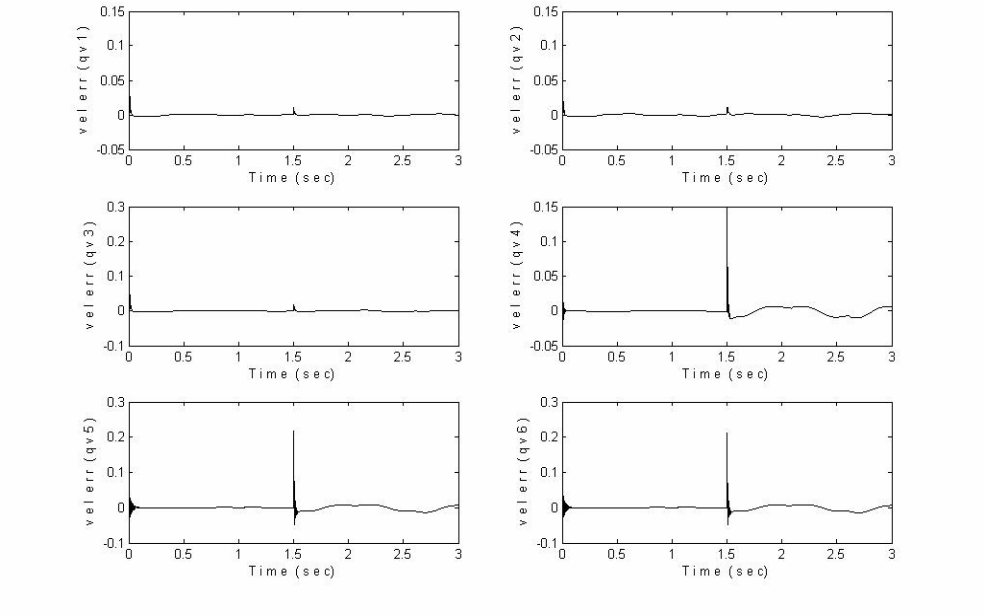

Fig 7. Velocity tracking errors in the case with disturbances

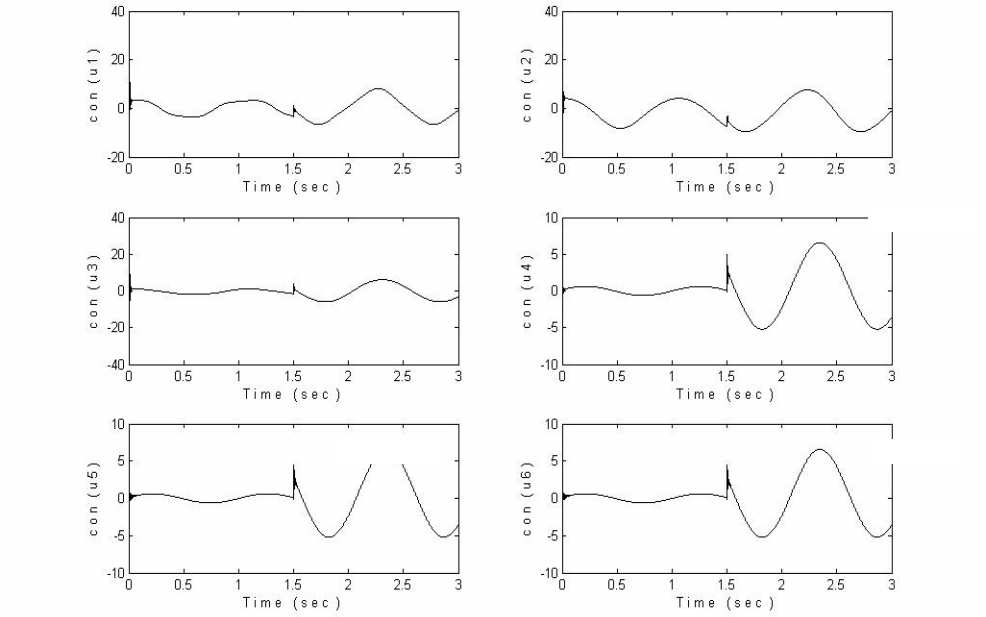

Fig 8 Profiles of the torques in the case with disturbances

In addition, it demonstrates robustness in performance against adverse effects such as structured and unstructured uncertainties. The designed structure and its simulation both verify that proposed controller provides a practical solution to the dilemmas faced by conventional control systems, because it shows better tracking performance even in the

presence of disturbances and the nonlinear effects due to the coupling between articulations. In addition, it demonstrates robustness in performance against adverse effects such as structured and unstructured uncertainties (e.g., robot inertia, coriolis effect, friction, and gravity). These effects influence the velocity, position and acceleration of the robotic joints and thus negatively impact the controller's performance and the life span of the manipulator itself. The designed neural-fuzzy system is understandable, flexible, and adaptive.

REFERENCES

[1] W. Wiliam, "Neurofuzzy Control of Modular and Reconfigurable Robots", IEEE Transactions on Mechatronics, Vol. 8, N^0. 3, September 2003.

[2] J. Meng, and G. Yang, "Robust Adaptive Control of Robot Manipulators Using Generalized Fuzzy Neural Networks", IEEE Transactions on Industrial Electronics, Vol. 50, N^o. 3, June 2003.

[3] M. Joo, and Y. Gao, "Robust Adaptive Control of Robot Using Generalized Fuzzy Neural Networks", IEEE Transactions on Industrial Electronics, Vol. 50, No. 3, June 2003, pp 620-628.

[4] S. Labiod, H. Chekireb , and S. Boucherit, « Commande Floue Adaptative Indirecte des Robots Manipulateurs », CNP'98 1er Colloque National sur la Productique 30-31 Mai 1998 Tizi-Ouzou, Algérie. pp 321-325.

[5] Yan-Qing, and K. Abrahim, "Compensatory Neurofuzzy Systems With Fast Learning Algorithms", IEEE Transactions On Neural Networks, Vol. 9, N^0 1, January 1998 pp 83-105.

[6] L. Peng, and W. Peng-Yung, "Neural-Fuzzy Control system for Robotic Manipulators", IEEE Control Systems Magazine, February 2002 pp. 53-63.

[7] M. Guihard, « Etude de lois de commande adaptatives d'actionneurs pneumatiques pour le contrôle dynamique d'un robot marcheur », Doctoral diss Pierre and Marie Curie Univercity, June 1995.

[8] B. Armstrong, O. Khetib, J.Burdick, "The explicit Dynamic Model and Inertial parameters of the PUMA560 Arm", IEEE 1986 pp. 510-518.

[9] R. Mellah, R. Toumi, « Commande neuro-floue du robot Puma560 muni de moteurs à courant continu dans les deux espaces tâche et articulaire », JESA .Vol 39 n°5-6 / 2005 pp739 - 765 . ISBN 2-7462-281-1 Hermès- Lavoisier Editeurs

[10] T. Lianfang, and C. Curtus, "Adaptive neuro-fuzzy control of a flexible manipulator", Mechatronics 15 (2005) pp 1305 - 1320.

[11] J. Meng, "Robust Adaptive Control of Robot Manipulators Using Generalized Fuzzy Neural Networks", IEEE on Industrial Electronics, Vol. 50, N^0. 3, June 2003.

[12] R. Mellah, and R. Toumi, « Modélisation et Commande d'un Bras Manipulateur muni de Vérins Pneumatiques », Conf. Intern. JTEA/IEEE - JTEA'98 - Nov 1998 Nabeul , - Tunisie

[13] R. Mellah, and R. Toumi, "Nonlinear Control of Robot PUMA 560 By Hybride Step by Step Motors". Fifth Intern, Symposium on Methods and Models in Automation and Robotics. MMAR'98 - August 1998 ,Miçdzyzdroje , Poland.

[14] R. Mellah, and R. Toumi, « Commande neuronale adaptative décentralisée du robot PUMA560 », SNAS'99 –Symposium National sur l'Automatique et les Systèmes .Nov 1999 - Annaba Algérie

[15] R. Mellah, and R. Toumi, "Decentralised adaptive neurofuzzy control with fast learning algorithms of robot manipulators". - Int. Conf. IASTED Robotics and Applications - RA 2003 - Juin 2003 -Salzburg (Autriche).

[16] R. Mellah and R. Toumi, « Commande Neuro-Floue par Mode Glissant du Robot PUMA 650 Muni de Vérins Pneumatiques », Conférence Internationale sur la Productique Alger, Octobre 2003.

[17] Y. Jin, "Decentralized Adaptive Fuzzy Control of Robot Manipulators", IEEE Transactions on Systems, Man, and Cybernetics – Part B : Cybernetics, Vol. 28, N^0 1, February 1998.

[18] J. Noriege, and H. Wang, "A direct adaptive Neural- Network Control for Unknown Nonlinear Systems and Its Application", IEEE Transactions on Neural Networks, Vol. 9, N^0 1, Jannuary 1998, pp 27 – 34.

[19] R. Jyh-Shing, T. Chuen, and S. Sai, "Neuro-Fuzzy Modeling and Control", Procedings of the IEEE, Vol. 83, N^0 3, March 1995 pp 378-404.

[20] K. Narendra, J. Balakrishnan, and M. Ciliz, "Adaptation and learning using multiple models, switching, and tuning", IEEE Control Systems, Vol. 14, n° 3, p. 37-53, 1995.

[21] C. Pam, "Intelligent control of stepping motor drive using an adaptive neuro-fuzzy inference systems", Information Sciences170 (2005) pp133-151.

[22] Y. Sahim, "Adaptive robust neural controller for robots", Robotics and Autonomous Systems 46 (2004) pp 175 - 184.

[23] C. Liang-Hsuan, and C. Cheng-Hsiung, "New Approach To Intelligent Control Systems With Self – Exploring Process", IEEE Transactions On Systems, Man, And Cybernetics, Vol. 33, N^o 1, February 2001, pp 56 – 66.

[24] J. Dominguez-lopez , and R. Damper, "Crowder RM, Harris CJ. Adaptive neuro-fuzzy control of a robotic gripper with on-line machine learning", Robotics and Autonomous Systems 48 (2004) pp 93 - 110.

[25] A. Ertugrul, and E. Murat, "A simulation comparison of intelligent control algorithms on a direct drive manipulator", Robotics and Autonomous Systems 49 (2004) pp 235-244.